Rapid Model Development Environment (ReMeDEe)

DRC created the PC-based Rapid Model Development Environment (ReMeDEe) to speed up the process of building logistics models and provide a user friendly model_execution environment that supports all of the MIL-STD-1388-1A quantitative analysis tasks. ReMeDEe combines the flexibility and immediacy of a spreadsheet with the power and maintainability of a general purpose programming language. It is particularly well suited to performing "what-if" analysis, sensitivity analysis, and constrained optimization during the course of support system trade studies.

ReMeDEe operating on an IBM PC provides an extremely cost effective approach for developing, tailoring, delivering and utilizing logistics models needed to perform all quantitative Logistics Support Analysis (LSA) tasks. It also provides users with a series of logistics models that all have the same "look and feel" with consistent model control via menu command lines, spreadsheet-like I/O forms, and graphical/tabular output. This significantly reduces the training required to learn a new model and use it with confidence. Models developed in ReMeDEe can be easily integrated with external systems to provide context sensitive on-line help, direct transfer of results to desktop publishing systems, and archiving of case study data via a database management system.

Model Development Aspects

ReMeDEe supports a rapid prototyping approach to model development using Borland International's Turbo PASCAL developers environment. It has the fastest compiler operating on the IBM PC, a great debugger, and several toolboxes for graphics, simple spreadsheets, a database management system, and libraries of numerical analysis routines all of which can be used in developing new models.

The ReMeDEe shell provides model developers with a common framework for quickly building a series of modularized logistics models by re-using segments of existing or newly developed models. ReMeDEe provides two options for optimum re-use of existing model code when developing new models. These options are either merging sections of code from prior models or chaining existing models by using the outputs of one model to feed the inputs of another. Both of these approaches allow the developer to rapidly assemble a model with all the functions needed to perform a particular study.

Our major technical innovation in creating the ReMeDEe shell was to adapt tools from an existing screen generator program written in Turbo PASCAL to build the user interface automatically. ReMeDEe does this for any model by driving the screen generator with a parser that uses information contained in the model variable declarations. Automatic generation of all I/O screens eliminates the time consuming effort needed to create customized user interfaces thereby promoting the rapid model development process.

ReMeDEe Architecture

ReMeDEe is divided into two major portions, the logistics model and the ReMeDEe shell (Figure 1). Models are written in Turbo PASCAL with the model equations in one file and the variable declarations augmented with additional information in another file (MGLOBALS.PAS). The ReMeDEe shell operates in two phases. In the first phase the model and its associated variables are compiled by the Turbo Pascal compiler, and an executable code is produced. In the second phase, the executable code automatically generates the I/O screens after reading the MGLOBALS.PAS file.

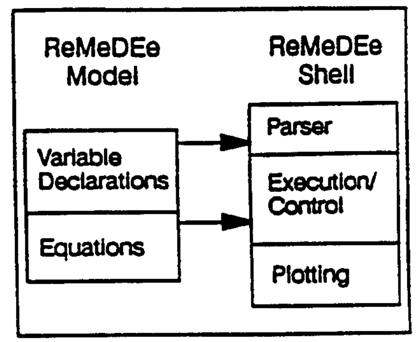


Figure 1. Relationship between the ReMeDEe Model and the ReMeDEe Shell

ReMeDEe Model Execution Mode

The model execution mode of the environment is natural and easy to use ("user friendly") providing the full functionality of "what if" and sensitivity analyses, plotting to various media, and the storage/retrieval of case history data. The ReMeDEe model execution environment provides users with a complete view of all model variables in a spreadsheet-like format. This enables total control over model execution, promoting flexible "what if" analysis and in depth diagnosis of model behavior. The "Select Set" function allows the user to focus his/her attention on a few key input/output variables that emerge during an investigation by collecting them on a single screen. Repetitive model execution is provided by the automated sensitivity analysis modes for either one or two independent input parameters with results displayed as two dimensional plots, a family of curves, or a three dimensional solid figure to visualize the output sensitivity surface. "Group plots" are used to show the coordinated sensitivity of several outputs due to changes in a particular, single, input. ReMeDEe

DRC ® DYNAMICS RESEARCH CORPORATION

includes a model execution mode that determines and displays the sets of pairs of independent input parameters (locus of points) corresponding to a desired model output which is very helpful while performing system trade studies. This mode can also be used to perform non-linear constrained optimization, for example, by picking the minimum life cycle cost solution along the locus of points that satisfy the system's availability goal. The results of performing sensitivity analyses and tradeoff studies can be incorporated directly into briefing charts and reports via ReMeDEe interfaces to both IBM PC (Lotus 1-2-3 and Enable) and Macintosh (Cricket Graph) desktop publishing systems. ReMeDEe provides the ability to save and retrieve all parameter files for case studies allowing users to return to previously considered supportability options while exploring the decision space in order to optimize system cost and availability.

Integration with a Database Management System

ReMeDEe models may be integrated with Borland's Paradox DBMS using PASCAL calls to the Paradox Engine. This integration provides users with the ability to archive, retreive and review case histories using predefined reports to investigate and compare baseline cases with improvement options. Model users, and not just developers, are able to create additional customized reports and charts using the Paradox Application Language without having to modify the ReMeDEe logistics models. Another advantage to our approach of using the DBMS is the ability to integrate independent models via a common database repository of parameters to ensure that correct data values are used in all models throughout the course of a study. It also provides an additional approach to integrating several independent models without extensively modifying their code.

Hypertext On-line Help System

Our approach to building the context sensitive on-line help system is to first create a ReMeDEe User Manual and Logistics Model Technical Manual and then develop a hypertext version of these on-line manuals. We next integrate this hypertext manual with the ReMeDEe model execution environment. Users with questions about the meaning and definition of model inputs and outputs appearing on ReMeDEe screens can instantly branch to explanatory articles and browse through related articles until their questions are satisfactorily answered. The on-line help system has been most useful to new and infrequent model users, giving them a sense of confidence in generating results using ReMeDEe logistics models.

ReMeDEe Applications

More than twenty models have been developed in ReMeDEe to answer a broad range of logistics support questions for a variety of application areas (Table 1). Many front-end, top-level, and detailed specific models for both Space-based and Air/Ground-based elements of the Strategic Defense System were created to support the MIL-STD-1388-1A LSA process. These models handle a variety of complex failure mechanisms, support system options, and repair opportunities governed by the dynamics of orbital mechanics. MODPALS was developed by using ReMeDEe's model chaining capability to integrate three other contractor models (for orbital mechanics/availability, life cycle cost, and combat effectiveness) originally written in BASIC and FORTRAN. The LAMP model was developed to compute the five R&M 2000 goals and is based on the integration of sections of code from five prior related models. Commonly used models that have been rehosted in ReMeDEe include the Logistics Analysis Model (LOGAM) developed by the United States Army Missile Command and the Cost Analysis and Strategy Assessment (CASA) model developed by the Defense Systems Management College.

Table 1. Example of ReMeDEe Models and their Logistics Capabilities

Models \ Support Analysis	Reliability/ Maintainability	Supply Support (Sparing)	Repair Level Evaluations	Manpower Personnel & Training	Influence Design	LCC	Cost/ Benefit Analysis
SDIO:							
Space/FEA	X	-	-	X	X	X	X
Space/TLA	X	-	-	X	X	X	X
AIR/FEA	X	X	X	X	X	X	X
AIR/AGTO	X	X	X	X	X	X	X
AIR/TLA	X	X	X	X	X	X	X
MODPALS	X	X	X	X	X	X	X
DAB2/ SASSAM	X	-	X	-	X	X	X
SAMSEM	X	-	X	-	X	X	X
Other:							
LAMP	X	X	-	X	X	X	X
FOG-M	X	X	-	X	X	X	X
MITS	X	X	-	X	X	X	X
LOGAM	X	X	X	X	-	X	X
CASA	X	X	X	X	-	X	X

Accession Number: 6006

Publication Date: Jan 01, 1996

Title: Rapid Model Development Environment (ReMeDEe)

Personal Author: Kleinman, A.; Szeto, K.

Corporate Author Or Publisher: Dynamics Research Corp., Andover, MA

Descriptors, Keywords: ReMeDEe Rapid Model Development Environment Logistics

Pages: 00010

Cataloged Date: Jan 16, 1996

Document Type: HC

Number of Copies In Library: 000001

Record ID: 40625

ReMeDEe - A RAPID MODEL DEVELOPMENT ENVIRONMENT FOR LOGISTICS ANALYSTS

Allan Kleinman and Katie Szeto

Dynamics Research Corporation (DRC)

ABSTRACT

DRC created the PC-based Rapid Model Development Environment (ReMeDEe) to speed up the process of building logistics models and provide a user friendly model execution environment that supports all of the MIL-STD-1388-1A quantitative analysis tasks. In this paper we describe ReMeDEe, explain how new models are developed and existing ones are ported to it, and show how to use the ReMeDEe/LOGAM model to perform a support system trade study. We also provide a description of ReMeDEe/LOGAM's context sensitive on-line help system consisting of hypertext versions of the ReMeDEe User Manual and LOGAM Technical Manual directly linked to the model execution environment.

INTRODUCTION

The requirements for ReMeDEe emerged from our logistics modeling activity which supported the LSA tasks we performed for the Strategic Defense Initiative Organization (SDIO). We had to develop many new models to handle a variety of space-based and ground-based systems with complex failure mechanisms, support system options, and repair opportunities governed by the dynamics of orbital mechanics. These models had to be developed quickly, so we chose a rapid prototyping approach with model developers working closely with logistics engineers. Our initial attempts at using spreadsheets (LOTUS 1-2-3) soon became unwieldy as the code became difficult to read and maintain, leading us to conclude that we needed to use a high level programming language. This decision was further supported by the requirement to re-use, enhance and integrate model code developed by other SDIO contractors, preferably within a common operating environment. We also had to provide a capability that allowed knowledgeable user analysts to modify models developed by us and other contractors.

The model execution mode of the environment had to be natural and easy to use ("user friendly") providing the full functionality of "what if" and sensitivity analyses, plotting to various media, and the storage/retrieval of case history data. From prior related modeling projects we knew that we couldn't afford the excessive amount of time spent on designing and implementing customized user interfaces and that we had to develop an automated I/O screen generator to build them. A further design constraint was that the environment had to be hosted on an IBM PC compatible computer which was commonly used by all contractor and government logistics analysts, thereby providing the most cost effective means for delivering logistics modeling capability to a wide variety of users.

Our approach to meeting the above requirements began with selecting a high level language commonly used by logistics analysts for building models as the basis for our environment. This limited our choices to FORTRAN, BASIC and PASCAL. We selected PASCAL because it is a strongly typed language that requires modelers to declare all variables used in the program, can be learned in about a day for those with any other computer language experience and has several libraries of numerical routines that can be used in developing new models. We also decided to use Borland International's Turbo PASCAL developer's environment. It has the fastest compiler operating on the IBM PC, a great debugger, and several toolboxes for graphics, simple spreadsheets, and a database management system.

Our major technical innovation in creating the ReMeDEe shell was to adapt tools from an existing screen generator program written in Turbo PASCAL to build the user interface automatically. ReMeDEe does this for any model by driving the screen generator with a parser that uses information contained in the model variable declarations. The ReMeDEe shell thus provides model developers with a common framework for quickly building a series of modularized logistics models by re-using segments of existing or newly developed models. It eliminates the time consuming effort needed to create customized user interfaces. It also provides users with a

U6006

series of logistics models that all have the same "look and feel" with consistent model control via menu command lines, spreadsheet-like I/O forms, and graphical/tabular output. This significantly reduces the training required to learn a new model and use it with confidence.

ReMeDEe OVERVIEW

ReMeDEe is divided into two major portions, the logistics model and the ReMeDEe shell (Figure 1). Models are written in Turbo PASCAL with the model equations in one file and the variable declarations augmented with additional information in another file (MGLOBALS.PAS). The ReMeDEe shell operates in two phases. In the first phase the model and its associated variables are compiled by the Turbo Pascal compiler, and an executable code is produced. In the second phase, the executable code automatically generates the I/O screens after reading the MGLOBALS.PAS file. ReMeDEe provides two options for optimum re-use of existing model code when developing new models. These options are either merging sections of code from prior models or chaining existing models by using the outputs of one model to feed the inputs of another.

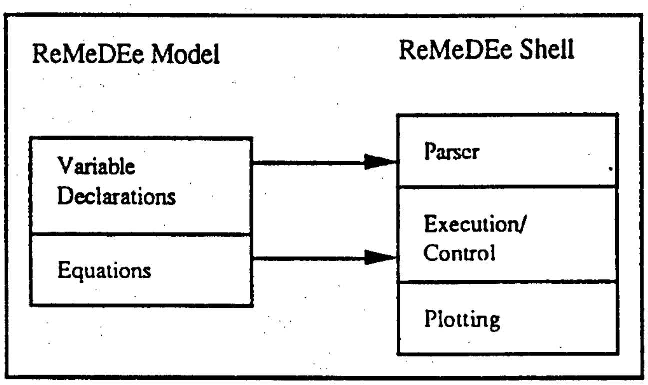


Figure 1. Relationship between the ReMeDEe Model and the ReMeDEe Shell

The ReMeDEe model execution environment provides users with a complete view of all model variables in a spreadsheet-like format. This enables total control over model execution, promoting flexible "what if" analysis and in depth diagnosis of model behavior. The "Select Set" function allows the user to focus his/her attention on a few key input/output variables that emerge during an investigation by collecting them on a single screen. Repetitive model execution is provided by the automated sensitivity analysis modes for either one or two independent input parameters with results displayed as two dimensional plots, a family of curves, or a three dimensional solid figure to visualize the output sensitivity surface. ReMeDEe includes a model execution mode that determines and displays the sets of pairs of independent input parameters (locus of points) corresponding to a desired model output which is very helpful while performing system trade studies. This mode can also be used to perform non-linear constrained optimization, for example, by picking the minimum life cycle cost solution along the locus of points that satisfy the system's availability goal. The results of performing sensitivity analyses and tradeoff studies can be incorporated directly into briefing charts and reports via ReMeDEe interfaces to both IBM PC (Lotus 1-2-3 and Enable) and Macintosh (Cricket Graph) desktop publishing systems. The ReMeDEe shell consists of 275 modules containing 12,400 source lines of code.

More than twenty models have been developed in ReMeDEe to answer a broad range of logistics support questions for a variety of application areas (Table 1). Many front-end, top-level, and detailed specific models for both Space-based and Air/Ground-based elements of the Strategic Defense System were created to support the MIL-STD-1388-1A LSA process. MODPALS was developed by using ReMeDEe's model chaining capability to integrate three other contractor models (for orbital mechanics/availability, life cycle cost, and combat effectiveness) originally written in BASIC and FORTRAN. The LAMP model was developed to compute the five R&M 2000 goals and is based on the integration of sections of code from five prior related models. Commonly used models that have been rehosted in ReMeDEe include the Logistics Analysis Model (LOGAM) developed by the United States Army Missile Command and the Cost Analysis and Strategy Assessment (CASA) model developed by the Defense Systems Management College. LOGAM will be used as an example throughout the rest of this paper to demonstrate ReMeDEe's model execution, model development, and hypertext on-line help capabilities.

ReMeDEe/LOGAM APPLICATION

LOGAM (Ref 1,2) is one of the Materiel Readiness Support Activity (MRSA) approved models for performing Army supportability trade studies in accordance with MIL-STD-1388-1A. It computes inherent and operational availabilities, provisioning requirements, manpower requirements, test equipment requirements, and life cycle cost. Table 2 gives examples of some of LOGAM's outputs and lists several of the generic system and LRU specific inputs. The model contains over 320 variables and was originally developed in

Table 1. Example of ReMeDEe Models and their Logistics Capabilities

Models \ Support Analysis	Reliability/ Maintain-ability	Supply Support (Sparing)	Repair Level Evaluations	Manpower Personnel & Training	Influence Design	LCC	Cost/ Benefit Analysis
SDIO:							
Space/FEA	X	–	–	X	X	X	X
Space/TLA	X	–	–	X	X	X	X
AIR/FEA	X	X	X	X	X	X	X
AIR/AGTO	X	X	X	X	X	X	X
AIR/TLA	X	X	X	X	X	X	X
MODPALS	X	X	X	X	X	X	X
DAB2/ SASSAM	X	–	X	–	X	X	X
SAMSEM	X	–	X	–	X	X	X
Other:							
LAMP	X	X	–	X	X	X	X
FOG-M	X	X	–	X	X	X	X
MITS	X	X	–	X	X	X	X
LOGAM	X	X	X	X	–	X	X
CASA	X	X	X	X	–	X	X

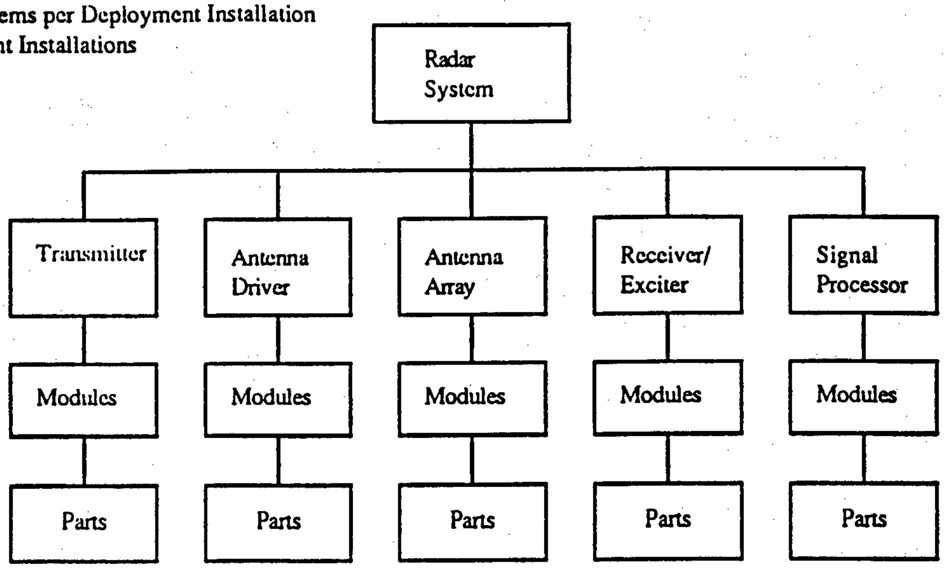
Table 2. LOGAM Systemwide and LRU Specific Inputs and Model Outputs

SYSTEMWIDE INPUTS	INDIVIDUAL LRU INPUTS	MODEL OUTPUTS
Manpower Labor Rates Training Inputs Shipping Inputs Storage Inputs Life Cycle (Years) Factory Lead Times Equipment Operating Time Factors Manpower Productivity Factors Work Weeks	LRU, Module, Part Costs Reliability (Failure Rate) MTTR Test & Repair Times Physical Characteristics (Weight-Cube) Modification Work Order Inputs Other Equipment Identifying Data	Design Supportability Life Cycle Cost O & M Cost Support Operational Readiness Availability Manpower Requirements (Maint & Support) Provisioning Requirements Test & Support Equipment Requirements & Costs Training Costs

FORTRAN for batch execution on a VAX computer. We converted this model to Turbo PASCAL and ported it to the ReMeDEe environment to significantly enhance its usability. An example of ReMeDEe/LOGAM's functionality will now be given for a Radar system performance improvement tradeoff study.

The Radar system LRU structure and input data (consisting of unit cost, MTBF, failure rate, and operating time fraction) are shown in Figure 2. The baseline system results (Table 3) from LOGAM show a life cycle cost of $536.21 million and a system operational availability of 0.853 that we would like to improve to 0.90. Table 3 also shows that the high driver LRU is the signal processor (Ao = .899) with all other LRUs having an Ao greater than .985. Options for improving the availability of the signal processor include a modification program for reducing its failure rate and the introduction of active redundant units in the design. Figure 3(a) is

- 24 Radar Systems per Deployment Installation
- 10 Deployment Installations



	Transmitter	Antenna Driver	Antenna Array	Receiver/ Exciter	Signal Processor
Avg LRU Cost	$161,926	$145,180	$60,620	$80,689	$119,300
Avg Module Cost	$ 24,290	$ 29,197	$24,240	$ 8,070	$ 13,015
Avg Part Cost	$ 467	$ 839	$ 420	$ 676	$ 907
MTBF (Hrs)	3,448	2,381	2,381	2,273	303
Failure Rate	0.00029	0.00042	0.00042	0.00044	0.0033
Oper Time Fraction	0.006	0.006	0.006	0.006	0.006

Figure 2. Radar System LRU Structure and Input Data

Table 3. Baseline Radar System LCC and Ao

LCCDAPAM = 5.3621E+08
OSTotCt = 3.9715E+08
SyAvailO = 0.852563

LRU Type	AvailO
Transmitter	0.990239
Antenna Driver	0.985928
Antenna Array	0.985623
Receiver/Exciter	0.985270
Signal Processor	0.899240

a 2D sensitivity plot showing the impact of reducing the failure rate from .0033 to .0011 on system operational availability and indicates the failure rate required to achieve Ao = .90. Figure 3(b) is a 3D sensitivity "family of curves" plot showing the combined effects of reducing the failure rate and adding either one or two redundant units and indicates the combination of failure rates and redundancy needed to meet the availability goal. Figure 3(c) shows the same results as a 3D solid figure and Figure 3(d) is a table of Ao achieved for each combination. Figure 3(e) shows the locus of points for combinations of failure rate and levels of redundancy required to meet the Ao = .90 goal. Figure 3(f) is a group plot showing the impact of reduced failure rate on both system availability and life cycle cost. Note that one of the limitations of LOGAM is that it does not include a cost estimating relation (CER) that generates the Research and Development (R&D) and Investment costs corresponding to implementing a failure rate reduction program. Therefore, we are unable to determine the minimum life cycle cost tradeoff of failure rate reduction versus level of redundancy options within ReMeDEe/LOGAM and would have to augment the model with an appropriate CER. The following description of our model installation process will show how easy it is to create and modify any given model.

MODEL INSTALLATION PROCESS

Models are created in ReMeDEe using the two phase approach described earlier in the ReMeDEe Overview. The model equation source code in Turbo PASCAL can either be independently derived, translated from existing FORTRAN or BASIC models, or most

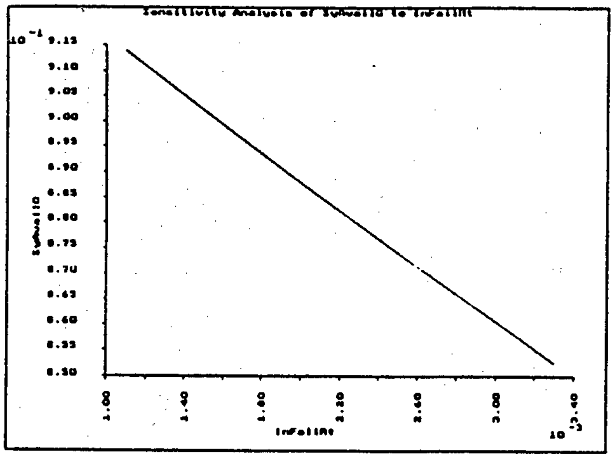


Figure 3(a). Example of a 2D Plot

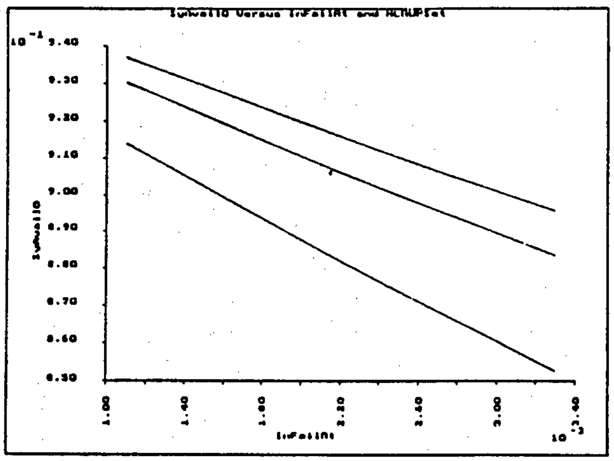


Figure 3(b). Example of Family of Curves

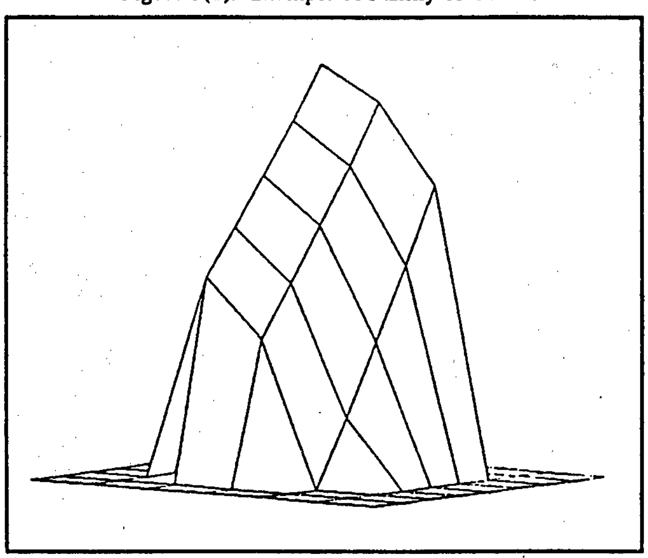


Figure 3(c). Example of a 3D Solid Figure

Sensitivity Analysis of SyAvailO

SyAvailO, System (Deployment Installation) Operational Availability

InFailRt, Input LRU Failure Rate (#/Hr: = 1/Baseline MTBF)

NLRUPSet, # Of LRUs Per LRU Set For Each LRU Type

	InFailRt	NLRUPSet 1.000	2.000	3.000
1	1.1000E-03	0.913951	0.930344	0.937113
2	1.6500E-03	0.897790	0.918078	0.926583
3	2.2000E-03	0.882191	0.906042	0.915811
4	2.7500E-03	0.867174	0.894502	0.905461
5	3.3000E-03	0.852563	0.883476	0.895669

Figure 3(d). Example of a 3D Plot Data Table

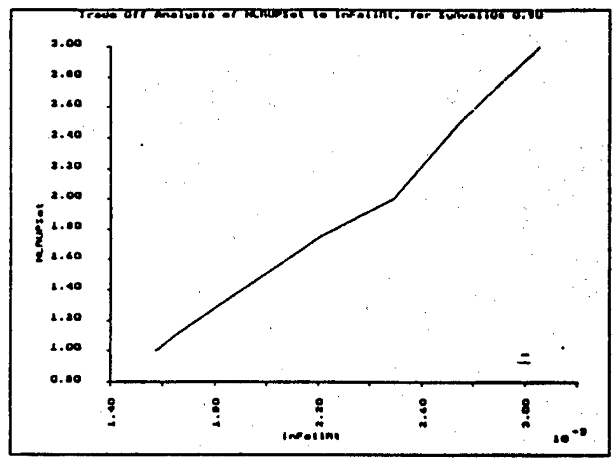


Figure 3(e). Example of a Tradeoff Study Plot

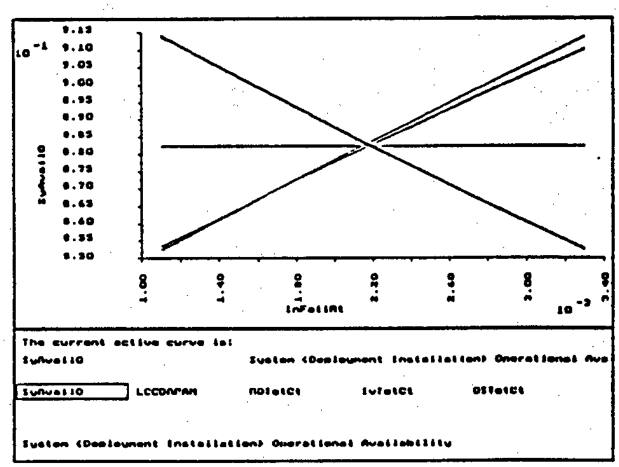


Figure 3(f). Example of a Group Plot

Figure 3. ReMcDEe/LOGAM Outputs for the Radar Example

easily extracted from a set of prior models already developed in ReMeDEe. In creating ReMeDEe/LOGAM we translated a rather large FORTRAN program and only made minimal changes. These include correcting several minor bugs, compressing the number of lines of code by using arrays where appropriate, and using longer, more meaningful, mnemonic variable names.

The LOGAM Technical/Programmer Manual (Ref 3) contains the description and derivation of all variables and equations used in the model. Figure 4(a) shows an excerpt describing the computation of the fraction of LRUs scrapped, repaired, or in limbo at the equipment, direct support, and general support maintenance levels. Figure 4(b) shows the corresponding portion of the FORTRAN code taken from the VAX listing. Figure 4(c) shows these same equations and section of FORTRAN code in their PASCAL version. Note the legend which shows the longer PASCAL variable names and the use of indices to handle the three maintenance levels with a single set of equations. For example, the variable FrURepr[IxML] represents the fraction of LRUs that are repairable at each maintenance level and is much easier to remember than the corresponding FORTRAN variables of FUE, FUO, and FUI. This is especially important when having to deal with over 320 model variables, many of which may be infrequently changed by the user.

```
USE=SUE+SUEC*FUEC
URE=SUEC*FUE
ULE=SUEC*FUEC
USO=SUO+SUOC*FUOC
URO=SUOC*FUO
ULO=SUOC*FUOC
USI=SUI+SUIC*FUIC
URI=SUIC*FUI
ULI=SUIC*FUIC
```

These statements compute the fraction of LRUs scrapped, the fraction of LRUs repaired and the fraction of LRUs in limbo at the Equipment, Direct Support and General Support levels, respectively.

Figure 4(a). LOGAM Technical Manual

```
C SCRAP, REPAIR, IN LIMBO FRACTIONS
      USE=SUE+SUEC*FUEC
      URE=SUEC*FUE
      ULE=SUEC*FUEC
      USO=SUO+SUOC*FUOC
      URO=SUOC*FUO
      ULO=SUOC*FUOC
      USI=SUI+SUIC*FUIC
      URI=SUIC*FUI
      ULI=SUIC*FUIC
```

Figure 4(b). LOGAM FORTRAN Code

```
for IxL:=1 to NTypeLRU do
begin      {of for IxL}
   for IxML:=Organization to Depot do
   begin      {of for IxML}
      TFrURepr[IxL,IxML]:=(1.0-FrUScrp[IxL,IxML])*
         FrURepr[IxL,IxML];
      TFrUScrp[IxL,IxML]:=FrUScrp[IxL,IxML]*(1.0-
         FrUScrp[IxL,IxML])*(1.0-FrURepr[IxL,IxML]);
      if (IxML<>Depot)
      then TFrULmbo[IxL,IxML]:=(1.0-
            FrUScrp[IxL,IxML])*
            (1.0-FrURepr[IxL,IxML])
      else TFrULmbo[IxL,IxML]:=0.0;
```

LEGEND:

USx → TFrUScrp[IxML]
URx → TFrURepr[IxML]
ULx → TFrULmbo[IxML]
SUxC → (1-FrUScrp[IxML])
FUxC → (1-FrURepr[IxML])
x = E,O,I → IxML = 1,2,3

Figure 4(c). PASCAL Version of LOGAM

Figure 4. LOGAM to ReMeDEe Conversion

ReMeDEe model variable declarations are collected in a separate file (MGLOBALS.PAS) and augmented with additional information that is decoded by the parser and used in the automated screen generator. Figure 5(a) shows an excerpt of the MGLOBALS.PAS file for the variables FrURepr and FrUScrp (fractions of LRUs scrapped). The LML corresponds to a modeler-defined variable type that sets aside a 10 by 4 array of reals (to be indexed by LRU and maintenance level). The information in the curly brackets from left to right is explained as follows: the first three numbers are the variable's maximum, minimum, and default 'values; the next three are the display format type, field size, and number of digits after the decimal point; the last item is a 60 character or less description of the variable which is displayed to the user during model parameter entry and output review as a helpful reminder of the variable's meaning. Figure 5(b) shows the screens that ReMeDEe generated from the information given in Figure 5(a). A much more in depth guide for the new or infrequent model user is provided by ReMeDEe's on-line help system.

FrURepr	: LML;	{1,0,0,1,6,4,LRU Repair Fraction}
FrUScrp	: LML;	{1,0,0,1,6,4,LRU Scrap Fraction}

Figure 5(a). MGLOBALS.PAS

FrURepr	LRU Repair Fraction			
	MLType			
LRUType	1	2	3	4
1	1.0000	1.0000	1.0000	1.0000
2	1.0000	1.0000	1.0000	1.0000
3	1.0000	1.0000	1.0000	1.0000
4	1.0000	1.0000	1.0000	1.0000
5	1.0000	1.0000	1.0000	1.0000
6	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000

FrUScrp	LRU Scrap Fraction			
	MLType			
LRUType	1	2	3	4
1	0.0000	0.0000	0.0000	0.0100
2	0.0000	0.0000	0.0000	0.0100
3	0.0000	0.0000	0.0000	0.0100
4	0.0000	0.0000	0.0000	0.0100
5	0.0000	0.0000	0.0000	0.0100
6	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000

Figure 5(b). ReMeDEe Screens

Figure 5. ReMeDEe/LOGAM Variable Declaration and Automatically Generated Screens

HYPERTEXT ON-LINE HELP

We first published a hardcopy version of a combined ReMeDEe User Manual and LOGAM Technical Manual (Ref 4), but several users who didn't enjoy thumbing through a thick book on their laps while operating ReMeDEe/LOGAM suggested that an on-line version would be much more helpful. In response, we decided to develop a hypertext version of the on-line manual and use this as a first stage in building a context sensitive on-line help system for ReMeDEe/LOGAM. A hypertext document can be considered as an electronic book consisting of a collection of articles (Ref 5, 6). These may be read linearly (in sequence) as an ordinary book, but more usefully "browsed" in a non-linear manner by a reader with a specific question. Hypertext links highlighted in the article being read provide the means of jumping to related articles that may be more pertinent. We selected the Hyperties system for the IBM PC from Cognetics Corporation to develop our on-line help system because of its ease of use in the browse mode and the simplicity of using its authoring system in developing hypertext documents. All we had to do was extract material from our ReMeDEe/LOGAM manual to create the articles, and carefully insert meaningful links between them. Figure 6(a) shows an example of an article with a highlighted link to a related article and the ability to return to the originating article. The Hyperties Index Screen mode provides an alphabetical listing of article titles which we utilized to great advantage by making all ReMeDEe functions and key LOGAM variables appear as article titles (Figure 6(b)). Hyperties also provides a Text Search mode (Figure 6(c)) where the reader creates a search string (containing boolean and/or conjunctions if desired) resulting in a list of articles matching the specified query. For example, the search string "failure rate" was identified in twelve articles, and is highlighted in the one selected for reading (Figure 6(d)).

We next integrated this hypertext manual with the ReMeDEe model execution environment to create our context sensitive on-line help system. This provides the user who has a question about some item on the ReMeDEe screen with the ability to switch from the model execution mode to the exact same place in the on-line manual where the explanation, along with related information provided by the hypertext system, can be found. We accomplished this ability to instantly switch back and forth between ReMeDEe and the manual by having them both co-resident in main memory using the terminate and stay resident

(TSR) feature. In order to create the screens in the hypertext manual, we first used a screen capture utility (in the graphics mode) and then superimposed a text screen containing the highlighted hypertext links that we built automatically using internal ReMeDEe screen generator files. The on-line help system has to be manually created for each model and is therefore best developed as the model changes from its prototype to production version.

Figure 6(a) shows an article describing the ReMeDEe R(etrieve) function that was obtained when a user switched to the help system prior to selecting the R(etrieve) function. While in the help system, the user can examine a typical response to executing the R(etrieve) function by following the highlighted link to the example in a related article. In a similar manner, users with questions about the meaning and definition of model inputs and outputs appearing on ReMeDEe screens can instantly branch to explanatory articles and browse through related articles until their questions are satisfactorily answered. The on-line help system has been most useful to new and infrequent model users, giving them a sense of confidence in generating results using ReMeDEe/LOGAM.

CONCLUSIONS

ReMeDEe provides an efficient, powerful, flexible and effective environment on an IBM PC for developing, tailoring, and utilizing logistics models in support of MIL-STD-1388-1A LSA tasks. Its consistent user interface and on-line help system significantly reduce training and the possibility of errors, especially when dealing with many logistics models. We are currently in the process of interfacing ReMeDEe with a commercially available database management system in order to integrate sets of related models through a common database leading to even greater productivity gains for model developers and users. We look forward to utilizing ReMeDEe as a vehicle for delivering the anticipated benefits from the recent dramatic

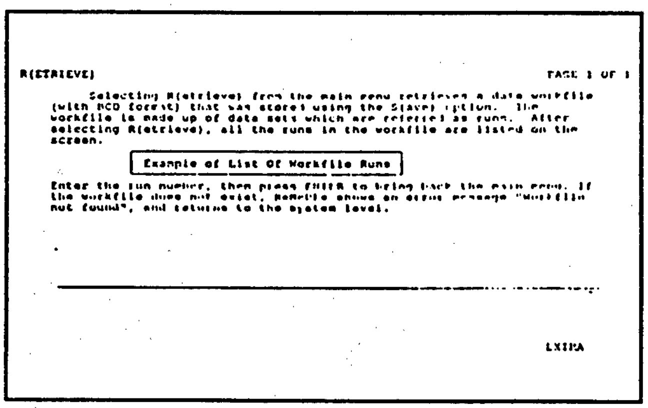


Figure 6(a). Example of an Article Describing the R(etrieve) Function

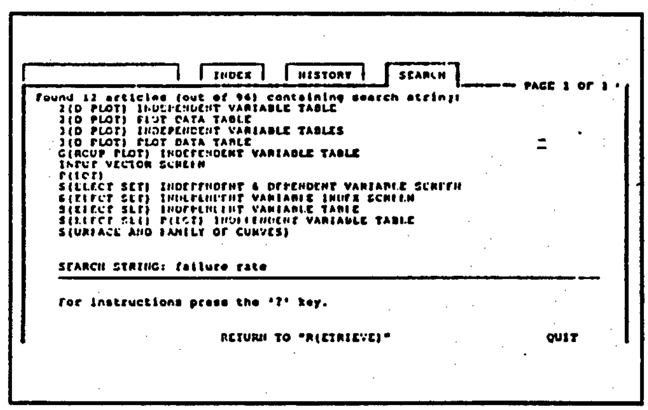


Figure 6(c). Example of a List of Articles Containing the Searched String

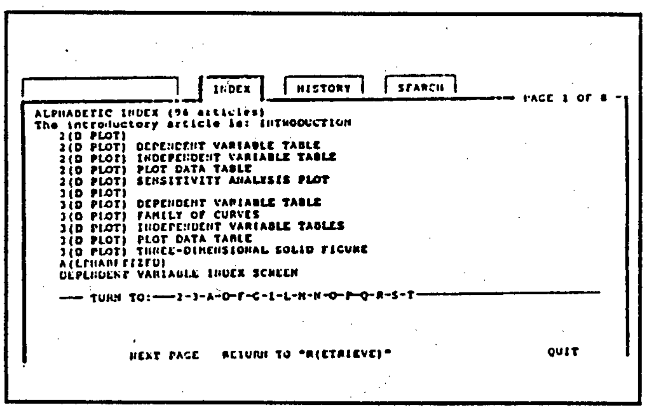


Figure 6(b). Example of an Index Screen

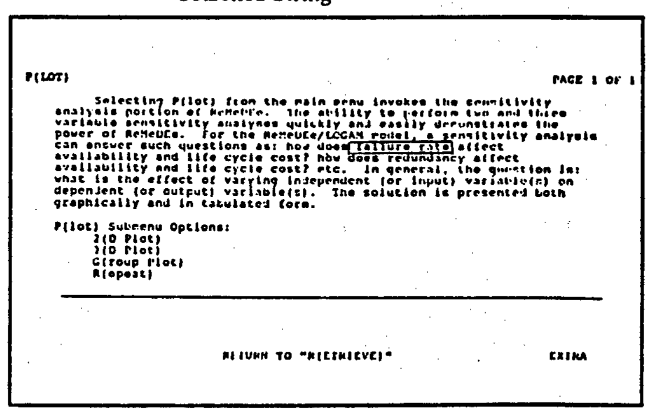


Figure 6(d). Example of an Article Containing the Searched String

Figure 6. ReMeDEe/LOGAM's On-Line Help System

improvements in PC hardware, operating systems, and software to the general logistics model user community.

ACKNOWLEGEMENTS

We would like to thank Pedro Restrepo, Jim Miller and Dave Welsh who conceived ReMeDEe and brought it to life. We also would like to thank Alan Leary at SDIO and Jim Pierce at USASDC for their continued encouragement which made ReMeDEe/LOGAM possible.

REFERENCES

1. R. Dotson and E. Seaberg, "LOGAM Executive Summary", U.S. Army Missile Command, Report D-82-2, Volume I, February 1980.

2. J. Wasson, "Logistic Support Analyses Modeling Techniques", Proceedings of the Society of Logistics Engineers 24th Annual Symposium - 1989.

3. R. Dotson, "LOGAM Technical/Programmer Manual", U.S. Army Missile Command, Report D-84-2, Volume III, February, 1985.

4. K. Szeto, "ReMeDEe/LOGAM User's Manual", Dynamics Research Corporation, September, 1990.

5. R. Peschke and K. Austin, "Hypertext: The Logistician's Answer to Information Overload?", Proceedings of the Society of Logistics Engineers 23rd Annual Symposium - 1988.

6. E. Conklin, "Hypertext: An Introduction and Survey", IEEE Computer, (September, 1987), pp. 17-41.

BIOGRAPHIES

Allan Kleinman is the manager of the Operations Research Section of the Logistics Systems Development Group at Dynamics Research Corporation. He is responsible for development and validation of logistics models used in performing system supportability studies. Mr. Kleinman has over ten years experience in developing logistic Decision Support Systems and has managed logistics support analyses for the US Army, Air Force and SDIO. He recently managed the development and use of DRC's Rapid Model Development Environment (ReMeDEe) for SDIO and contributed to the development of key logistics models embedded within the Air Force Weapon System Management Information System (WSMIS). Mr. Kleinman received a BSEE from the Polytechnic Institute of Brooklyn and a MSEE from Brown University.

Katie Szeto is a senior systems analyst in the Operations Research Section of the Logistics Systems Development Group at Dynamics Research Corporation. She is responsible for developing and using logistics models to perform system supportability analyses for SDI systems. Ms. Szeto recently verified and validated the US Army Missile Command's LOGAM model, rehosted it from a mainframe to DRC's PC-based Rapid Model Development Environment (ReMeDEe) and integrated it with a hypertext version of the user's manual and theory manual to provide on-line context sensitive help. She received a BS and MS in Chemical Engineering from Tufts University and a MS Engineering (Electrical) from Northeastern University.